

Area-Feature Boundary Labeling¹

MICHAEL A. BEKOS¹, MICHAEL KAUFMANN², KATERINA POTIKA¹
AND ANTONIOS SYMVONIS^{1,*}

¹*School of Applied Mathematical & Physical Sciences, National Technical University of Athens,
15780 Zografou, Athens, Greece*

²*Institute for Informatics, University of Tübingen, Sand 13, 72076 Tübingen, Germany*

*Corresponding author: symvonis@math.ntua.gr

Boundary labeling is a relatively new labeling method. It can be useful in automating the production of technical drawings and medical drawings, where it is common to explain certain parts of the drawing with text labels, arranged on its boundary so that other parts of the drawing are not obscured. In boundary labeling, we are given a rectangle R which encloses a set of n sites. Each site s is associated with an axis-parallel rectangular label l_s . The labels must be placed in distinct positions on the boundary of R and they must be connected to their corresponding sites with polygonal lines, called *leaders*, so that the labels are pairwise disjoint and the leaders do not intersect each other. In this paper, we study a version of the boundary labeling problem where the sites can ‘float’ within a polygonal region. We present a polynomial time algorithm, which runs in $O(n^3)$ time and produces a labeling of minimum total leader length for labels of uniform size placed in fixed positions on the boundary of rectangle R .

Keywords: boundary labeling; map labeling; design and analysis of algorithms

Received 12 January 2008; revised 11 June 2009

Handling editor: Suchi Bhandarkar

1. MOTIVATION

Placing extra information, in the form of text labels, next to features of a drawing is an important task in the process of information visualization. Usually, it is desired that the label placement is done so that each label is close to the feature (*site*) it describes and is not overlapping with any other label. In general, it is NP-hard to obtain optimal label placements under this assumption [2]. Wolff and Strijk [3] maintain an extensive bibliography about map labeling.

In medical and technical drawings, a commonly used approach is to explain certain features of the drawing with text labels that are arranged on its boundary. This is illustrated in Fig. 1, which depicts a medical drawing of a human heart. It is easy to see that if the labels were placed close to the features they describe, they would obscure the underlying drawing and they would possibly overlap with each other. Besides, there are cases, e.g. when the labels are very large or the features are too many, where it is impossible to find a label placement so that the labels are close to the features they describe. To cope with

such cases, Bekos *et al.* [4] focussed on *boundary labeling* and were the first to algorithmically study this labeling approach.

In this paper, we study a version of the boundary labeling problem where sites can ‘float’ within a region. This is motivated by the fact that we often want to label *area features* of a drawing, e.g. a body part in a medical drawing or a machine part in a technical drawing. Figure 2 illustrates the anatomy of the human thigh muscles. Observe that each muscle occupies a region within the figure. According to our approach, instead of arbitrarily selecting a point to represent each of these regions, we associate them with polygonal areas (refer to the dashed rectangular areas of Fig. 2), so that any site inside (or on the boundary of) these polygonal areas can be selected to represent the corresponding region.

To the best of our knowledge this is the first time where boundary labeling is applied to area features. The proposed algorithm uses similar techniques (i.e. bipartite matching and crossing elimination) as the corresponding one for the case of point-site labeling [4]. The main difference from the previous work is the addition of an extra step regarding the computation of the shortest leader between a particular site and its associated label (see Section 5.1). Also, the crossing elimination procedure uses quite different and more advanced techniques in order

¹This article is based on the preliminary version [1].

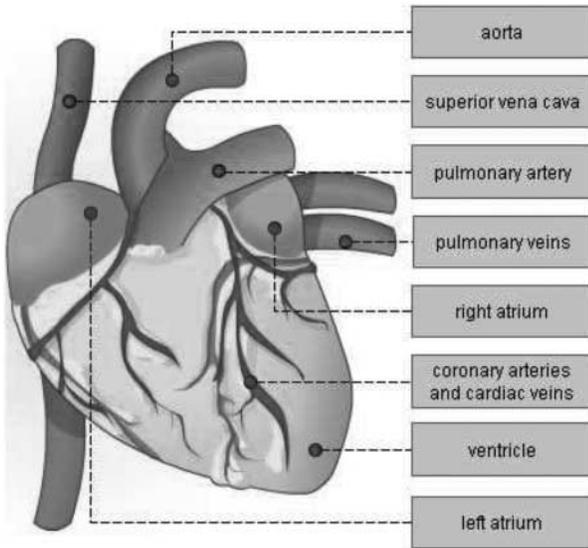


FIGURE 1. Illustration of a human heart.

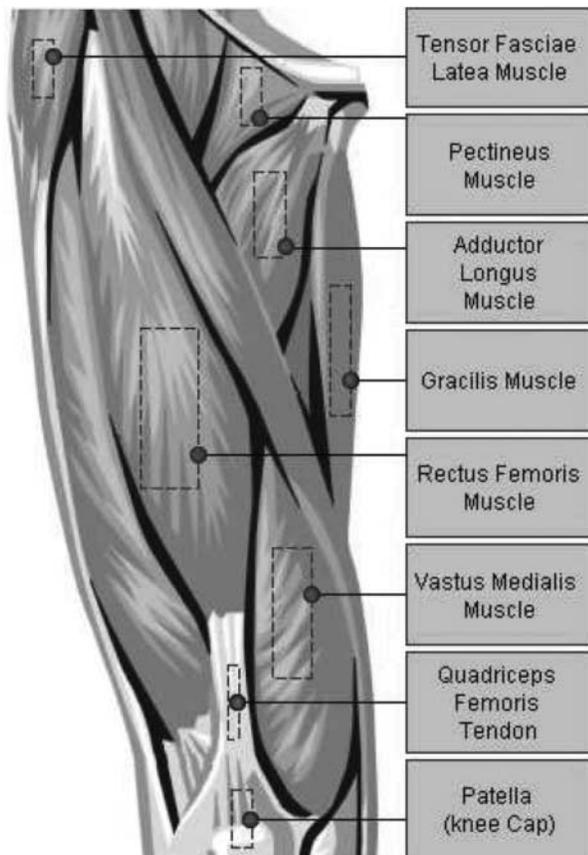


FIGURE 2. Illustration of the human thigh muscles.

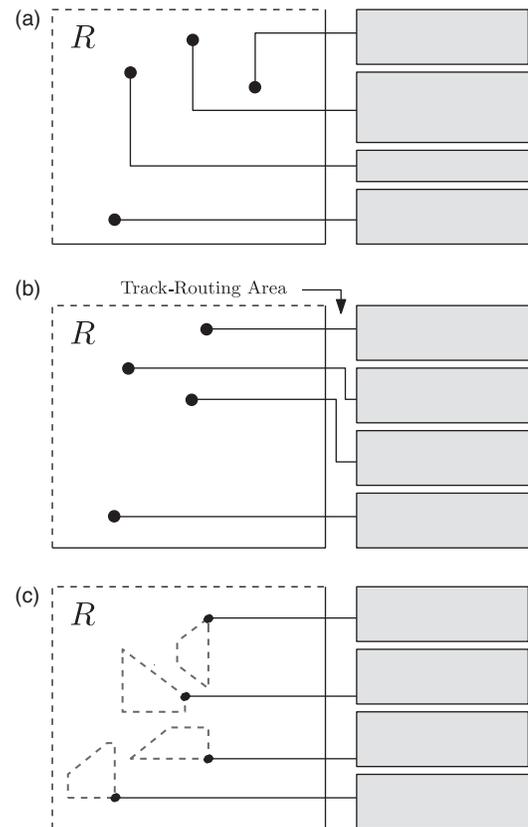


FIGURE 3. Different boundary labeling models: (a) leaders of type po , (b) leaders of type opo and (c) polygonal sites.

to obtain the desired labeling. Unfortunately, in the case of area-sites, we cannot use Vaidya's algorithm [5] to reduce the time complexity of the proposed algorithm, as in the corresponding case of point-sites [4], since the underlying graph is not geometric.

This paper is structured as follows. In Section 2, we formally define the area-feature boundary labeling problem. Section 3 reviews previous results on boundary labeling. In Section 4, we present necessary notation and terminology. Section 5 studies the problem of minimizing the total leader length when type- opo leaders (i.e. three-segment rectilinear polygonal curves; see Fig. 3b) are used and the labels are placed on all four sides of R . In Section 6, we study the problem of minimizing the total leader length when type- po leaders (i.e. two-segment rectilinear polygonal curves; see Fig. 3a) are used and the labels are placed on two opposite sides of R . We further prove that a crossing-free labeling with such leaders is not always feasible. We conclude in Section 7 with open problems and future work.

2. PROBLEM DEFINITION

In boundary labeling, we are given a set P of n sites s_i , $i = 1, 2, \dots, n$, each associated with a rectangular label l_i of dimensions $w_i \times h_i$. The site set P and the underlying drawing

are enclosed in an axis-parallel rectangle R of sufficient size, which is called *enclosing rectangle*. The labels must be placed on distinct positions on the boundary of R so that they do not overlap, and must be connected to their corresponding sites with non-intersecting curves, called *leaders*. Such labelings are referred to as *legal* or *crossing-free boundary labelings*.

Given that several parameters (sites, labels, leaders, enclosing rectangle) are involved in boundary labeling, there exist several variations of boundary labeling, each giving rise to a different labeling model.

2.1. Sites

The *sites* model features of the drawing. In the simplest form of the problem, they model point locations in a drawing, e.g. a city center, or the capital of a prefecture. In this case, each site s_i is associated with a point $p_i = (x_i, y_i)$ in the plane (see Fig. 1 or Fig. 3a and b). To avoid leader overlaps, which would reduce the readability of the implied drawings, we make an additional assumption regarding the location of the sites as follows: we assume that the sites are placed in *general position*, i.e. no three sites are collinear and no two sites share the same x - or y -coordinate.

As mentioned in Section 1, in practice we often want to label area features, e.g. a region of a drawing. In ‘*area-feature boundary labeling*’ we are given as part of the input a set of regions in each of which a point-site has to be selected to represent the region and to be connected to its corresponding label. In order to be consistent with the terminology used in map labeling, we refer to these regions as *area-sites*, or simply as sites when the context is clear. We study the cases where the area-sites are either simple polygons, rectangles or line segments. Note that the ‘general position assumption’ that requires that *no two sites share the same x - or y -coordinate* will be extended to apply to area-sites. A detailed definition is given in Section 4.

2.2. Leaders

Sites are connected to their corresponding labels with non-intersecting polygonal lines, which are called *leaders*. We denote by c_i the leader of site s_i (see Fig. 4). Since we aim at simple and easy-to-understand labelings, we only consider leaders that consist of either a single axis-parallel straight-line

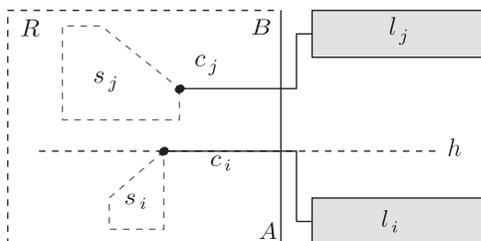


FIGURE 4. Leader c_i (c_j) is oriented toward (away from) corner A.

segment or a sequence of axis-parallel segments, each of which is either parallel (p) or orthogonal (o) to the side of R containing the label it leads to. The *type* of a leader is defined by an alternating string over the alphabet $\{p, o\}$. In our approach, we use leaders of type po and opo as follows:

Type- po leaders: Leaders of type po consist of two line segments. The first one is parallel (p) to the side of R containing the label it leads to, whereas the second one is orthogonal (o) to that side (see Fig. 3a). A degenerated case of a po -leader is a leader of type o , which consists of only one line segment orthogonal to the side of R containing the label it leads to.

Type- opo leaders: Following the same notation scheme, leaders of type opo consist of three line segments (see Fig. 3b). Again, leaders of type o are trivially considered to be of type opo , as well.

For each type- opo leader, we further assume that it has its parallel p -segment outside the enclosing rectangle R , routed in the so-called *track-routing area* (see Fig. 3b). This is done in order to avoid leader bends within the underlying drawing or intersections between leaders and polygons, in the case where we want to visualize the polygon that corresponds to each region of the drawing. Note that if we permit p -segments in the interior of R , then the boundary labeling problem seems to become harder. To realize that, observe that in the degenerated case where the p -segment of a type- opo leader coincides with the area-site (i.e. the first o -segment is of zero length) then this leader trivially becomes po , and therefore, the model in which we permit p -segments in the interior of R can be viewed as a mixed model, where both opo - and po -leaders are allowed. We further assume that the width of the track-routing area is fixed and large enough to accommodate all leaders with a sufficient distance. In fact, we consider a rectangle $R' \supset R$ broader than R and use $R' \setminus R$ as a fixed-width track-routing area, i.e. we place all leader segments that are parallel to the corresponding side of R in $R' \setminus R$. Since the track-routing area is fixed and every leader has to cross it in order to reach its label, the contribution of the width of the track-routing area to the total length of each leader is ignored.

Note that there exist several ways to determine the exact locations of the p -segments of the leaders within the track-routing area. For instance consider the case where the labels are placed on the right side of R . Then, the simplest way to determine the exact x -coordinates of the p -segments of the leaders is to assign their offset from the boundary according to the following rule: ‘*For downward leaders (oriented toward the bottom-right corner) use as offset the index of the leader’s label in a bottom-to-top ordering of the labels while, for upward leaders (oriented toward the top-right corner) use as offset the index of the leader’s label in a top-to-bottom ordering of the labels.*’ Thus, in the description of our algorithms we will focus on the labeling problem and ignore the leader routing within the track-routing area.

2.3. Labels

Each leader connects a site to a label and touches the label at a point on its side that faces the enclosing rectangle. This point is called *label-port*. We can assume either *fixed ports* where the leader is only allowed to use a fixed set of label-ports (a typical case is where the leader uses the middle point of the label side; see Fig. 3a and b) or *sliding ports* where the leader can touch any point of the label's side (see Fig. 3c).

The labels are placed on the boundary of an axis-parallel rectangle R of height H and width W the left-top corner of which coincides with the origin of our coordinate system.¹ In general, we assume that the labels are of arbitrary size (*non-uniform labels*), i.e. label l_i , which corresponds to site s_i , has height h_i and width w_i . However, several versions of the boundary labeling problem with non-uniform labels are hard to solve. To realize that, assume that labels of variable height must be placed either to the left or right side of the enclosing rectangle R , and that the label heights sum up to twice the height H of R . It is clear that the task of assigning the labels to the two sides corresponds to the PARTITION problem, which is a well-known NP-complete problem. Therefore, it is reasonable to separately consider the restricted cases where the labels are of *uniform size*, or of *maximum uniform size*, i.e. labels of uniform size covering the full length of the side of the rectangle in which they reside. This is also motivated by the fact that in real applications the labels often contain single line text, e.g. the name of a body part in a medical drawing or the name of a machine part in a technical drawing. Figure 3b and c display labelings with labels of uniform size, whereas Fig. 3a displays a labeling with non-uniform labels.

2.4. Optimization criteria

Our aim is to obtain legal labelings that optimize some criterion. Keeping in mind that we want to obtain simple and easy-to-understand labelings, we focus on the *total leader length minimization problem*. Of course, there exist several other aesthetic criteria, which can be adopted, mostly from the areas of VLSI and Graph Drawing, such as the minimization of the total number of leader bends or the minimization of the length of the longest leader.

3. PREVIOUS WORK

The first results on boundary labeling were presented by Bekos *et al.* [4]. A variety of models based on the type of the leaders, the location of the labels and the size of the labels were studied. The focus of their work was on efficient algorithms for minimizing the total leader length and for minimizing the total number of

leader bends. Bekos *et al.* [6] studied a variation of boundary labeling, where the labels are arranged in multiple stacks on one side of the rectangle. In a subsequent work, Bekos *et al.* [7] used a quite similar (to boundary labeling) method to label collinear sites, which may lie along a horizontal or sloped input line.

Benkert and Nöllenburg [8] presented algorithms for minimizing the total leader length with type-*po* or type-*do*² leaders, when uniform labels are allowed to be placed on one side of the enclosing rectangle. Recently, Benkert *et al.* [9] studied boundary labeling along a new line of research. They formulated the problem as an optimization problem, where the objective function is a general quality function which evaluates the niceness of the resulting labeling. Then, using dynamic programming, they presented several results for the case where the labels are of uniform size, placed on one side of the enclosing rectangle and the leaders are either of type *po* or of type *do*.

However, Benkert and Nöllenburg [8] and Benkert *et al.* [9], report that the production of a boundary labeling with leaders of type *do* is not always feasible. This motivated Bekos, Kaufmann, Nöllenburg and Symvonis to introduce two new types of octilinear leaders and they further proved that by combining them, the boundary labeling problem is always feasible [10]. Their main contribution was an algorithm for solving the total leader length minimization problem assuming labels of uniform size.

Kao *et al.* [11] introduced the *many-to-one boundary labeling* to describe a variation of boundary labeling, where several sites are associated with a common label. In the case of many-to-one boundary labeling, the presence of crossings among leaders often becomes inevitable. Therefore, they presented several algorithms, approximations and heuristics for minimizing the total number of crossings.

4. NOTATION AND TERMINOLOGY

In this section, we present some necessary notation and definitions that are heavily used in the remainder of the paper. We denote the number of area-sites (and consequently the number of labels) by n and the maximum number of corners of each area-site by k , where k is a constant.

Simple polygons: A simple polygon (or simply polygon) is represented by a set of k points (of integer coordinates), referred to as *corners*, indexed $\{1, \dots, k\}$ in the clockwise direction that define its boundary. In Fig. 4, polygon s_i has four corners, while polygon s_j has five corners. We extend the notion of general position for sites represented as points to polygons, as follows: we say that polygons s_1, s_2, \dots, s_n are in *general position* if for each pair of indices i, j with $i \neq j$, the following are satisfied:

- (i) there do not exist two corners belonging to polygons s_i and s_j with the same x - or y -coordinate;

²A *do*-leader consists of a diagonal followed by an orthogonal segment.

¹Actually, the labels are placed at the boundary of the larger rectangle R' that includes the track-routing area. Since we ignore the influence of the track-routing area, we treat the labels as being placed at the boundary of rectangle R .

- (ii) there do not exist two corners belonging to polygon s_i and label l_j with the same x - or y -coordinate.

In the rest of the paper, we assume that the set of area-sites to be labeled are always in general position.

Site-ports: Each leader that connects an area-site s_i to a label touches the boundary of s_i at a point. This point is referred to as *port of area-site* s_i or simply as *site-port of* s_i and is denoted by p_{s_i} . In Fig. 4, the ports of both area-sites s_i and s_j coincide with one of their corners.

Leader orientation: Consider a type-*opo* leader c_i that originates from site-port p_{s_i} of a polygon s_i and is connected to a label on the right side AB of the enclosing rectangle R . The line that contains the first *o*-segment of the leader c_i (i.e. the one which is incident to p_{s_i}) divides the plane into two half-planes (see the dashed line h of Fig. 4). We say that leader c_i is *oriented toward* corner A of the rectangle R if both corner A and the label-port of s_i are on the same half-plane, otherwise, we say that leader c_i is *oriented away* from corner A (see Fig. 4). In the case of a type-*o* leader, we consider the leader to be oriented toward corner A (and also toward corner B).

5. FOUR-SIDED AREA-FEATURE BOUNDARY LABELINGS WITH TYPE-OPO LEADERS

In this section, we study the area-feature boundary labeling problem with type-*opo* leaders. As already stated, we will restrict our presentation to the case where the area-sites are represented as simple polygons. We further assume that we have fixed labels of uniform size, placed on all four sides of rectangle R . We present a polynomial time algorithm, that returns a legal labeling of minimum total leader length.

Since the labels have uniform size, each area-site s_i can be connected to any label l_j . We seek to connect each area-site s_i to a label l_j and to specify two points, one on the boundary of s_i (site-port of s_i) and one on the boundary of l_j (label-port of l_j), so that the total leader length is minimized. Algorithm 5.1 outlines our approach.

Initially, we construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$, where $P = \{s_1, s_2, \dots, s_n\}$, $L = \{l_1, l_2, \dots, l_n\}$, $E = \{(s_i, l_j) \mid s_i \in P, l_j \in L\}$ and $w : E \rightarrow \mathbb{R}$ is a cost function (see step *A* of Algorithm 5.1). Each edge $e_{ij} = (s_i, l_j) \in E$ of G is assigned a weight $w(e_{ij}) = d_{ij}$, where d_{ij} is equal to the Manhattan length of a *shortest* (under the Manhattan metric) leader that connects area-site s_i to label l_j . We proceed by computing a minimum-cost bipartite matching on G , i.e. we compute a matching between area-sites and labels that minimizes the total Manhattan distance of the matched pairs (see step *B* of Algorithm 5.1). We can obtain a labeling M of minimum total leader length as follows. If an edge $e_{ij} = (s_i, l_j) \in E$ is selected in the matching, then we connect

area-site s_i to label l_j using a leader of length $w(e_{ij})$ (see step *C* of Algorithm 5.1). Labeling M is of minimum total leader length, but it might contain crossing leaders. However, we can eliminate all crossings and therefore obtain a legal labeling M' , keeping the total leader length unchanged, i.e. equal to that of M (see step *D* of Algorithm 5.1). In the following sections, we describe in detail each step of Algorithm 5.1.

5.1. Shortest leader computation

In Step *A* of Algorithm 5.1, we have to compute the minimum Manhattan distance between every area-site $s_i \in P$ and every label $l_j \in L$. This is equivalent to computing the shortest type-*opo* leader that connects area-site s to label l , for all pairs (s, l) , where $s \in P$ and $l \in L$.

LEMMA 5.1. *Let d be the length of a shortest (under the Manhattan metric) leader, say c , which connects area-site s to label l and assume that c originates from a side of area-site s with a slope ϕ that is not a multiple of 45° . Then, we cannot identify a leader c' of equal length (namely of length d), which connects s to l and originates from the same side of area-site s using a different site-port along this side.*

Algorithm 5.1: (Four-side area-*opo*)

input : A set $P = \{s_1, \dots, s_n\}$ of n area-sites in the plane and a set $L = \{l_1, \dots, l_n\}$ of n uniform labels placed on the boundary of R .

output: A crossing-free four-side type-*opo* labeling of minimum total leader length.

Step A. Shortest Leader Computation:

Construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$. The weight $w(e_{ij})$ of an edge $e_{ij} = (s_i, l_j) \in E$ is the Manhattan length of the *shortest* (under the Manhattan metric) leader, say d_{ij} , which connects s_i to l_j .

Step B. Compute Minimum-Cost Bipartite Matching:

Proceed by computing a minimum-cost perfect bipartite matching \mathcal{M} on G , i.e. compute a matching between area-sites and labels that minimizes the total Manhattan distance of the matched pairs.

Step C. Obtain a labeling M as follows:

foreach (edge $e_{ij} = (s_i, l_j) \in \mathcal{M}$) **do**
 Connect s_i to l_j with a leader of length $w(e_{ij})$

Step D. Eliminate crossings:

Eliminate all crossings of leaders and obtain a crossing-free labeling M' , keeping the total leader length unchanged, i.e. equal to that of M .

Proof. To prove this lemma, assume without loss of generality that label l is on the right side of R and that $0 < \phi < 45$. For the sake of contradiction, assume that there exists a leader $c' \neq c$ of length d that originates from a different site-port along the side of area-site s that hosts the site-port of leader c . Let p_s and p'_s be the ports of area-site s occupied by the leaders c and c' , respectively, and assume that $y(p'_s) > y(p_s)$, i.e. p'_s is above p_s . Then, by moving the site-port of leader c along the side of area-site s and toward p'_s , its length is reduced, which contradicts the fact that leader c is a shortest leader from s to l . Analogous reasoning yields the desired property in the case where $45 < \phi < 90$ or in the case where ϕ corresponds to a negative sloped side of area-site s . \square

From Lemma 5.1, it follows that the shortest leader from an area-site s to a label l is uniquely defined in the case where it has its port along a side of area-site s with a slope ϕ that is not a multiple of 45° . However, this property does not necessarily hold if slope ϕ is a multiple of 45° . An example is illustrated in Fig. 5, where the definition of the shortest leader that connects area-site s to label l is not unique. Therefore, in this case there exist several leaders of minimum leader length connecting s to l .

The following lemma demonstrates that it is always possible to identify a shortest leader that satisfies a specific property with respect to its origin and/or destination. This holds for both the case of the unique shortest leader attached to a specific side of an area-site with a slope that is not a multiple of 45° (as indicated by Lemma 5.1) as well as for the case of equivalent shortest leaders (i.e. shortest leaders that connect the same site to the same label) that are attached to the side of an area-site with a slope that is a multiple of 45° .

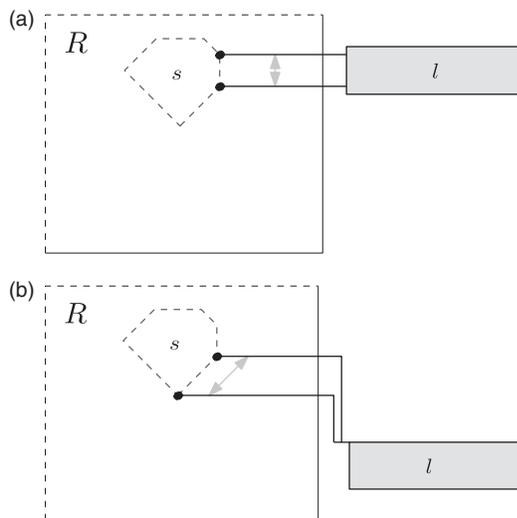


FIGURE 5. Illustration of different leaders of minimum length connecting s to l : (a) the leader connecting s to l is of type o and (b) the leader connecting s to l is of type opo .

LEMMA 5.2. Let d be the length of a shortest (under the Manhattan metric) leader, say c , which connects area-site s to label l . Then, we can identify a leader c' of equal length (namely of length d), which connects s to l for which exactly one of the following statements hold.

- (i) Leader c' is a type- opo leader with non-zero length p -segment originating from some corner of area-site s and leading to some corner of label l .
- (ii) Leader c' is of type- o and either originates from some corner of area-site s or leads to some corner of label l .

Proof. First observe that no leader c' can satisfy both statements of this lemma due to the fact that the leader in case (i) has a non-zero length p -segment while the leader in case (ii) has no p -segment at all. Also note that in case (ii) the leader c' cannot originate from some corner of area-site s and also lead to some corner of label l due to the general position requirement.

Without loss of generality, we assume that label l is placed at the right side of rectangle R . We consider two cases based on the slope of the side at which the site-port of leader c is located.

Case 1: The site-port of the shortest leader c is located at a side that is vertical, horizontal or diagonal (at angles which are multiples of 45° with respect to the axes). Consider first the subcase where the shortest leader c connecting area-site s to label l is an opo -leader with non-zero length p -segment. The fact that c is a shortest leader implies that it connects to a corner of label l . Without loss of generality, assume that it leads to the top-left corner of l and, thus, it originates from a site-port x above the horizontal strip occupied by label l . (The case where leader c leads to the bottom-left corner of l is treated symmetrically.) If c originates from a corner of area-site s , then c satisfies the conditions of statement (i). Thus, $c' = c$ and we are done. Hence, assume that c does not originate from a site-port x that is also a corner of s . This implies that x belongs to an edge of area-site s that is diagonal and has positive slope (see Fig. 6a). Note that if the edge was vertical, then the length of the leader could be shortened by moving x toward the bottom of the drawing, contradicting the assumption that c is the shortest. Then, x can be moved to the rightmost point of the edge without changing the total leader length. This new leader c' satisfies the conditions of statement (i).

Now consider the subcase where c is a type- o leader, i.e. a type- opo leader with a zero-length p -segment. If c originates from a corner of s or it leads to a corner of l , then it satisfies the conditions of statement (ii). Thus, $c' = c$ and we are done. Hence, assume that it originates from a non-corner point x of an edge of area-site s and it leads to a non-corner point y of the left side of label l . Also observe that x must lie on a vertical edge of area-site s . Note that if the edge was diagonal, then the length of the leader could be shortened by moving x toward its leftmost point, contradicting the assumption that c is the shortest. Then, by translating the type- o leader toward the top of the drawing and until we first either hit the top-left corner of the label (see

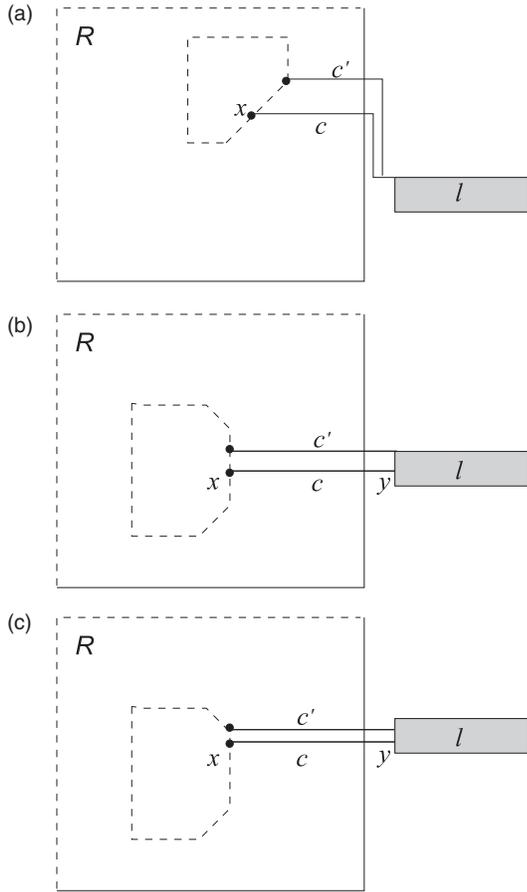


FIGURE 6. Illustration of the cases arising in the proof of Lemma 5.2: (a) a type-*opo* leader originating from a diagonal edge of the area-site; (b) a type-*o* leader that is translated until it meets the top-left corner of the label; and (c) a type-*o* leader that is translated until it meets the topmost point of the side-edge.

Fig. 6b) or the topmost point of the site-edge (see Fig. 6c), we obtain a new leader c' with length equal to that of c that satisfies the conditions of statement (ii).

Case 2: The site-port of the shortest leader c is located at a side with a slope that is not a multiple of 45° (with respect to the axes). By Lemma 5.1, we have established that the shortest leader c is unique and, thus, $c = c'$. Hence, we have to establish that leader c satisfies the property.

First consider the subcase where the shortest leader c connecting area-site s to label l is an *opo*-leader with non-zero length p -segment. The fact that c is a shortest leader implies that it connects to a corner of label l . If the site-port of leader c coincides with a corner of s , we are done. Otherwise, by moving the site-port of c along the side of s (in a direction that depends on the slope of the side and the location of the label relative to side), we can always shorten the length of the leader, which is a clear contradiction. Thus, the site-port of c always coincides with a corner of l and, thus, it satisfies statement (i).

Now consider the subcase where c is a type-*o* leader. If c leads to a corner of l or it originates from a corner of s , then it satisfies the condition of statement (ii). Otherwise, by moving the label-port of c toward one of its two label corners (and maintaining it as a type-*o* leader), we can always shorten its length, which is a clear contradiction. Thus, statement (ii) is satisfied. \square

From Lemma 5.2, it follows that the candidate site-ports for the leader that connects area-site s to label l are the corners of area-site s and the horizontal (or vertical) projections of the corners of label l to the sides of area-site s , if any. Similarly, the candidate label-ports for this leader are the corners of label l and the horizontal (or vertical) projections of the corners of area-site s to the side of label l that faces R , if any. Therefore, since we assumed that the area-sites are in general position, a computation of such leaders yields a solution containing no overlapping leaders, i.e. the first *o*-segments of two leaders cannot overlap.

Algorithm 5.2 outlines our method for computing the shortest type-*opo* leader that connects s to l , for all pairs (s, l) , where $s \in P$ and $l \in L$. Initially, for each area-site s_i we construct a set s_i^{port} that contains all candidate site-ports of s_i . Similarly, for each label l_j we construct a set l_j^{port} that contains all candidate label-ports of l_j (see Step B of Algorithm 5.2). We proceed by computing the Voronoi diagram H_i (see [12, p. 158, 13]) under the Manhattan distance for each set s_i^{port} . To compute the shortest (under the Manhattan metric) leader that connects area-site s_i to label l_j , we utilize the notion of the nearest neighbor in H_i . More precisely, for each point in l_j^{port} we determine its nearest neighbor in H_i and we compute their Manhattan distance. Then, the shortest leader that connects area-site s_i to label l_j corresponds to the one that minimizes the distances computed above, and we set $w(e_{ij})$ to be this distance.

THEOREM 5.1. *Algorithm 5.2 computes the minimum distance under the Manhattan metric between any label and any simple polygon, when the labels are placed in fixed positions on all four sides of rectangle R in $O(n^2 \log n)$ time.*

Proof. Step A of Algorithm 5.2 trivially needs $O(n^2)$ time. Each label l_j contributes at most $2k$ elements to each set s_i^{port} , where k is the maximum number of corners of each area-site. From those we only keep the two points closer to the label (one for each extension of the label that intersects the area-site). Therefore, the number of elements in each set s_i^{port} is $O(k + n)$. Similarly, each area-site s_i contributes at most k elements to each set l_j^{port} . From those we only need to keep the one that is closer to the label. This implies that the number of elements in each set l_j^{port} is $O(n)$. Thus, the total time needed for Step B of Algorithm 5.2 is $O(kn^2)$.

The construction of the Voronoi diagram H_i of Step C.1 of Algorithm 5.2 can be done in $O(k' \log k')$ time [13], where $k' = O(k + n)$. Finding the nearest neighbor of a point q in the Voronoi diagram H_i costs $O(\log k')$ time [13]. Therefore, we compute Step C.2 in a total of $O(kn \log k')$ time. Then, the

Algorithm 5.2: (Shortest leader computation)

input : A set $P = \{s_1, \dots, s_n\}$ of n area-sites on the plane and a set $L = \{l_1, \dots, l_n\}$ of n uniform labels placed on the boundary of R .

output: A complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$, where the weight $w(e_{ij})$ of edge $e_{ij} = (s_i, l_j) \in E$ is the length of the *shortest* (under the Manhattan metric) leader, which connects s_i to l_j .

Step A: Initiate output graph.

Construct a graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$, where the weight $w(e_{ij})$ of an edge $e_{ij} = (s_i, l_j) \in E$ is initially equal to zero.

Step B: Determine all possible site and label-ports.

for each area-site s_i ($1 \leq i \leq n$) **do**
 Add the corners of s_i to s_i^{port} .
for each label l_j ($1 \leq j \leq n$) **do**
if (l_j is located to the right or left side of R) **then**
 Find the intersection points of each edge of s_i with the horizontal lines passing from each corner of label l_j that faces R and in each case select the one closest to l_j .
else
 Find the intersection points of each edge of s_i with the vertical lines passing from each corner of label l_j that faces R and in each case select the one closest to l_j .
 Add these points to s_i^{port} .
for each label l_j ($1 \leq j \leq n$) **do**
 Add the corners of l_j that face R to l_j^{port} .
for each area-site s_i ($1 \leq i \leq n$) **do**
 Find the intersection points of the edge of l_j that faces R with the lines perpendicular to it passing from each corner of area-site s_i and in each case select the one closest to l_j .
 Add this point to l_j^{port} .

Step C: Shortest leader computation.

for each area-site s_i ($1 \leq i \leq n$) **do**
 (i) Construct the Voronoi diagram H_i (under the Manhattan distance) for s_i^{port} .
 (ii) **for** each element of l_j^{port} **do**
 Find the nearest neighbor in H_i and compute their Manhattan distance. Set $w(e_{ij})$ to be the minimum such distance.

running time of Algorithm 5.2 is $O(n(k' + n) \log k')$ and since we assume that k is constant, the running time of Algorithm 5.2 is $O(n^2 \log n)$. \square

5.2. Minimum-cost bipartite matching

In Step B of Algorithm 5.1, we have to compute a minimum-cost perfect bipartite matching on the output graph of Algorithm 5.2. This problem is also known as *assignment problem* and can be efficiently solved by means of the well-known Hungarian method in $O(n^3)$ time [14]. Note that the output graph of Algorithm 5.2 is not geometric. Therefore, we cannot make use of a faster algorithm to compute the minimum-cost matching, as the one by Vaidya [5].

5.3. Computation of a length-optimal labeling

In Step C of Algorithm 5.1, we compute a labeling M that is optimal in terms of total leader length, but it might contain crossings. This is done based on the computed matching of Step B of Algorithm 5.1. In fact, the matching defines a site-label assignment at each side of the enclosing rectangle R . Then, the routing of the p -segments of the leaders within the track-routing area must be done in such a way that no two leaders intersect or overlap each other. (A possible way to perform the routing of the leaders within the track-routing area was presented in Section 2.2.) Since the width of the track-routing area is fixed, the arbitrary distribution of the p -segments of the leaders within the track-routing area does not affect the total leader length. Obtaining a minimum total leader length labeling in a model that utilizes a variable width track-routing area is open.

5.4. Crossings elimination

In this section, we describe how to eliminate all crossings from labeling M (obtained from Step C of Algorithm 5.1). Note that labeling M is of minimum total leader length and we eliminate all crossings keeping the total leader length unchanged, i.e. optimal.

We observe that we can have two kinds of crossings: (i) crossings taking place inside the enclosing rectangle R and (ii) crossings taking place in the track-routing area. We proceed first to eliminate all crossings taking place inside R .

LEMMA 5.3. *Let M be a minimum-length opo-labeling (which might contain crossings) obtained from Step C of Algorithm 5.1. Let c_i and c_j be a pair of leaders that intersect inside R and originate from area-sites s_i and s_j , respectively. Then, the following statements hold.*

- (i) *The labels l_i and l_j of these leaders lie on two adjacent sides of the enclosing rectangle R . Let A be their incident corner.*
- (ii) *Leaders c_i and c_j are oriented toward corner A of the enclosing rectangle R .*

- (iii) Leaders c_i and c_j can be rerouted so that they do not cross each other and the sum of their leader lengths remains unchanged.

Proof. (i) First observe that labels l_i and l_j cannot both lie on the same side of R , since the area-sites are in general position and the crossings within the track-routing area are ignored. Note that the general position restriction implies that the first o -segments of the leaders do not overlap each other, which could lead to a degenerated case of crossing in the interior of R .

We now consider the case where the labels l_i and l_j lie on opposite sides of rectangle R . Then, since the leaders intersect each other, the segments of the leaders that are inside the rectangle (and incident to the area-sites) have to intersect. However, since these segments are parallel to each other, they have to overlap. Note that an overlap as the one depicted in Fig. 7a, where both leaders c_i and c_j originate from a corner of their corresponding area-sites, cannot occur because of the general position assumption. Therefore, from Lemma 5.2 it follows that the only case where such an overlap can occur is the one

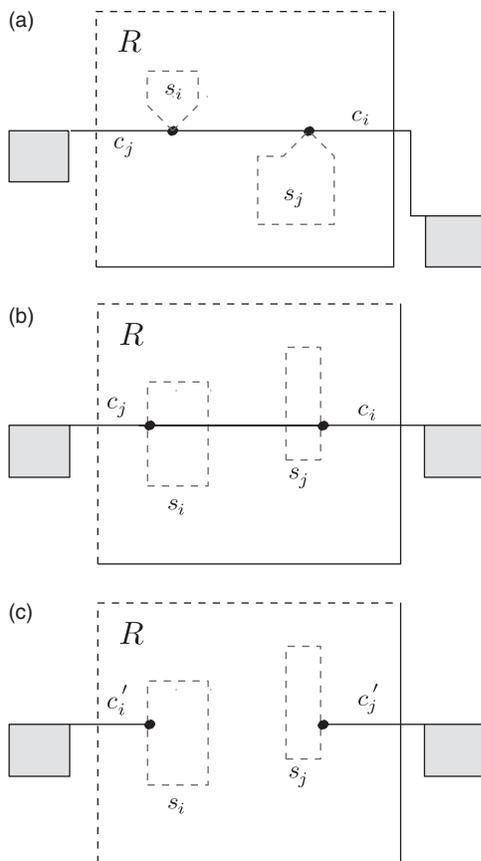


FIGURE 7. A pair of overlapping leaders c_i and c_j : (a) a case which cannot occur because of the general position assumption, (b) before rerouting and (c) after rerouting.

depicted in Fig. 7b, where (i) both leaders c_i and c_j are of type o , (ii) c_i and c_j lead to the corners of their associated labels and (iii) these corners have the same y -coordinate. Again, by swapping the labels to which each area-site is connected, we can eliminate the overlap (see Fig. 7c). However, the total leader length is reduced, since the overlapping parts of both leaders c_i and c_j are totally eliminated, which is a contradiction since we assumed that the total leader length of the labeling is minimum.

- (ii) Let A be the corner that is incident to the two sides of the rectangle R containing the labels associated with leaders c_i and c_j . In order to show that in a labeling of minimum total leader length both leaders c_i and c_j are oriented toward corner A , it is enough to show that (in a labeling of minimum total leader length) it is impossible to have one or both leaders oriented away from corner A . A detailed proof of a similar statement has been presented for the corresponding point-site labeling problem [4]. Hence, the detailed proof is omitted. The idea is to lead any combination of different leader orientations into a contradiction, because of reductions of the total leader length. Having eliminated the cases, where one or both crossing leaders are oriented away from corner A , the only case left is the one where both leaders c_i and c_j are oriented toward corner A . Such a case is depicted in Fig. 8a.
- (iii) In order to show that leaders c_i and c_j can be rerouted so that they do not cross each other and the sum of their leader lengths remains unchanged, we partition the first segment of each of the leaders c_i and c_j into two subsegments from their crossing point to the sides of

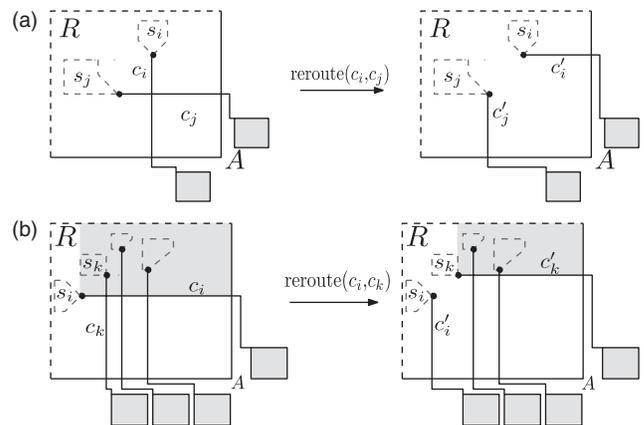


FIGURE 8. An illustration of the crossing elimination procedure: (a) in an *opo*-labeling of minimum total leader length, two crossing leaders that are oriented toward the corner incident to the sides of the rectangle containing their associated labels can be rerouted so that their crossing is eliminated and the sum of their leader length is unchanged and (b) rerouting the crossing leaders c_i and c_k , the leftmost area-site incident to an intersecting leader connected to a label on the right side of the rectangle is located to the right of area-site s_j .

the enclosing rectangle (see Fig. 8a). Then, obtain the new leaders c'_i and c'_j by sliding the (sub)segments of leaders c_i and c_j , leaving their sum unchanged. \square

LEMMA 5.4. *Let M be a minimum-length opo-labeling (which might contain crossings) obtained from Step C of Algorithm 5.1. Let c_i and c_j be a pair of leaders that intersect in the track-routing area and originate from area-sites s_i and s_j and lead to labels l_i and l_j , respectively. Then, the following statements hold.*

- (i) Leaders c_i and c_j have the same orientation.
- (ii) Leaders c_i and c_j can be rerouted so that they do not cross each other and the sum of their leader lengths remains unchanged.

Proof. (i) Since the crossing of the leaders c_i and c_j lies within the track-routing area, the labels l_i and l_j should be on the same side, say the right side AB , of the enclosing rectangle R (see Fig. 9). For the sake of contradiction, assume that there exist two crossing leaders c_i and c_j that are oriented toward corner B and corner A , respectively. This case is illustrated in Fig. 9a. By swapping the labels to which each area-site is connected, we can eliminate the crossing (see Fig. 9b). However, the total leader length is reduced. To see this, consider the projections of the p -segments of both c_i and c_j on the y -axis and observe that they share a common overlapping part (refer to the dashed line segment of Figure 9a). This part (which is of length greater than zero due to the general position requirement) is totally eliminated after the rerouting of the leaders c_i and c_j , which is a contradiction since we assumed that the total leader length of the labeling is minimum. Therefore, all

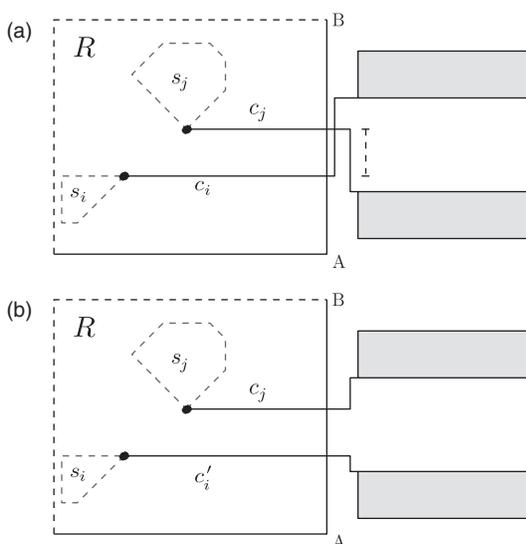


FIGURE 9. A pair of intersecting leaders c_i and c_j : (a) before rerouting and (b) after rerouting.

crossings within the track-routing area involve leaders with the same orientation.

- (ii) In order to show that leaders c_i and c_j can be rerouted so that they do not cross each other and the sum of their leader lengths remains unchanged, it is enough to swap the labels to which each area-site is connected. Then, it is easy to show that this procedure does not affect the total length contributed by the crossing leaders. Furthermore, the position of the first o -segment of both leaders remains unchanged after the rerouting process. \square

In the following, we show that given a labeling of minimum total leader length, which may contain crossings as those described in Lemmas 5.3 and 5.4, we can efficiently resolve all crossings yielding a new crossing-free labeling so that the total leader length is unchanged.

THEOREM 5.2. *Let M be an opo-labeling (which might contain crossings) obtained from Step C of Algorithm 5.1. We can always identify a crossing-free opo-labeling M' with total leader length equal to that of M (step D of Algorithm 5.1). Moreover, labeling M' can be obtained in $O(n \log n)$ time.*

Proof. This theorem can be proved in a similar fashion as the corresponding one for the point-site labeling problem [4]. Therefore, we will only sketch the basic ideas of this proof. The proposed algorithm acts in two independent phases. In the first phase, we eliminate all crossings taking place inside the enclosing rectangle R . In the second phase, we independently eliminate the crossings that take place in the track-routing area. Note that possible reroutings performed in order to eliminate crossings in the track-routing area cannot introduce crossings in the interior of R . Hence, in two independent phases we can eliminate all crossings, resulting in a new labeling M' without any crossings and of total leader length equal to that of M , i.e. of minimum total leader length.

Description of the first phase: In the first phase, we eliminate crossings taking place in the interior of R . Our method performs two passes over the area-sites, one in the left-to-right and one in the right-to-left direction. Consider the left-to-right pass. The right-to-left pass is treated symmetrically. In the left-to-right pass of labeling M , we consider all area-sites with labels on the right side of the rectangle. Assuming that the area-sites can be sorted based on the coordinates of their ports, we examine the area-sites from left to right and we are interested only in those that have leaders crossing in the interior of R . Let s_i be the leftmost such area-site and let c_i be the leader that connects it to its corresponding label on the right side of the rectangle (see the left drawing of Fig. 8b). Lemma 5.3(i) implies that leader c_i intersects only with leaders that are connected to labels either on the top or bottom side of rectangle R . Without loss of generality, assume that c_i is oriented toward the bottom-right corner of the rectangle, say A . Then all leaders that intersect c_i in the interior

of R have their labels on the bottom side of R and are also oriented toward A (by Lemma 5.3(ii)). Let c_k be the leftmost such leader, and let s_k be its incident area-site. According to Lemma 5.3(iii), we can reroute leaders c_i and c_k so that the total leader length remains unchanged (see the right drawing of Fig. 8b).

Continuing in the same manner, we can prove that the leftmost area-site that participates in a crossing in the interior of R (in the left-to-right pass) is pushed to the right, which guarantees that all ‘left-to-right’ crossings in the interior of R are eventually eliminated. Moreover, when the left-to-right pass is executed, it is impossible to introduce new ‘right-to-left’ crossings. Therefore, when the two independent passes over the site set are completed, we have eliminated all crossings in the interior of R .

Description of the second phase: Having eliminated the crossings taking place in the interior of R , we next proceed to eliminate all crossings in the track-routing area. First consider the area-sites that have to be connected to the labels attached to the right side of R . By Lemma 5.4, it follows that in order to avoid leader crossings within the track-routing area, the vertical order of the site-ports of those area-sites should be identical to the vertical order of their corresponding labels along the right side of R . Hence, all crossings within the track-routing area can be easily eliminated by reassigning the sites to the labels, so that the vertical order of the site-ports is identical to the

vertical order of their corresponding labels along the right side of R and this cannot affect the total leader length (because of Lemma 5.4(ii)).

In a straightforward approach, the first phase of Step D of Algorithm 5.1 needs $O(n^2)$ time. However, its time complexity can be further improved to $O(n \log n)$ following a similar approach as the one presented for the corresponding point-site labeling problem [4] by employing a data structure supporting queries of the form ‘given a set of points Q that change under insertions and deletions, a threshold value y_0 and a query range (l, r) , return the point of Q with the smallest x -coordinate that is located within the rectangle $(l, r) \times (y_0, H)$ ’. The MinXInRectangle query just described can be answered in $O(\log n)$ time by employing a dynamic priority search tree based on half-balanced trees [15, p. 209]. The second phase of Step D of Algorithm 5.1 trivially needs an extra cost of $O(n \log n)$ time in order to compute the proper routing of the leaders within the track-routing area. \square

THEOREM 5.3. *Given a label set L of n uniform-sized labels placed at fixed positions on the boundary of a rectangle R and a site set P of n area-sites in general position inside R , we can compute in $O(n^3)$ time a legal *opo*-labeling of minimum total leader length.*

Proof. In Step A of Algorithm 5.1, we construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all area-sites $s_i \in P$ and all labels $l_j \in L$. Each edge $(s_i, l_j) \in E$

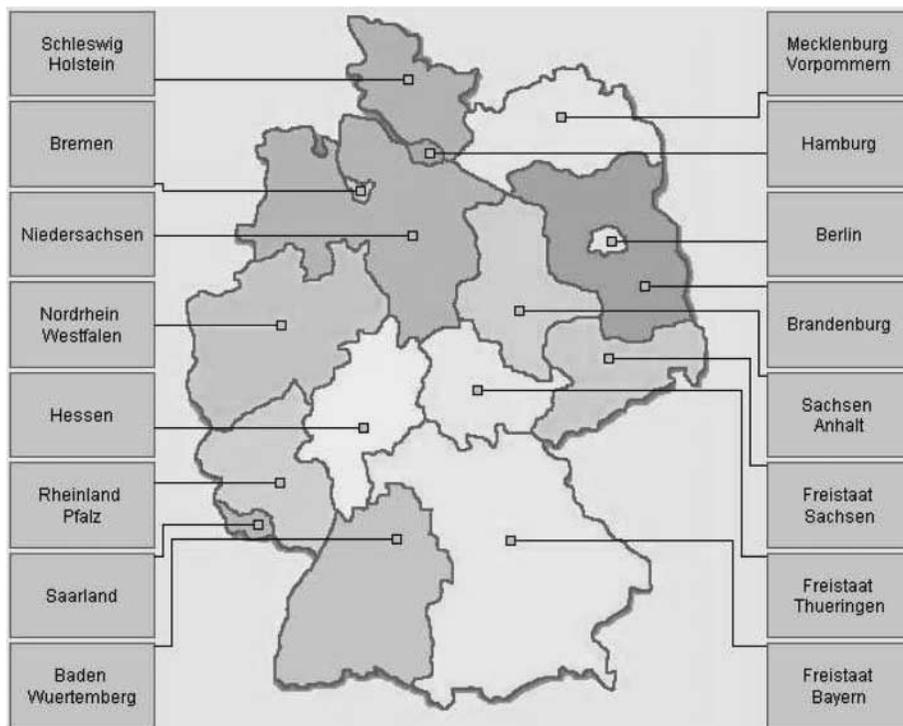


FIGURE 10. A labeling of a regional map of Germany with *opo*-leaders; a point is the representative of each region.

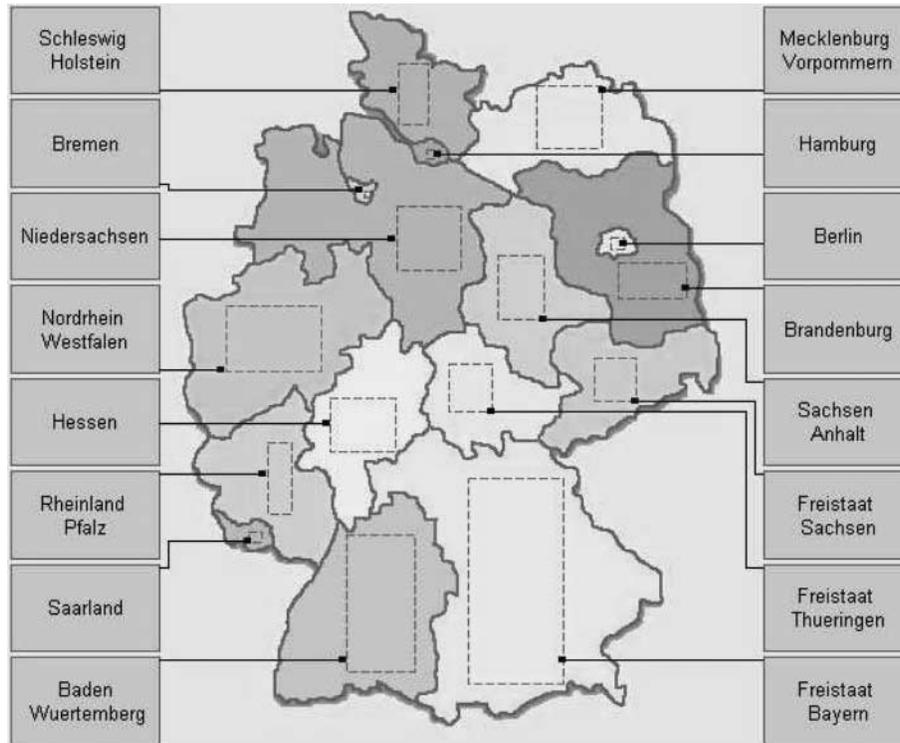


FIGURE 11. A visually improved map for the drawing of Fig. 10; each region is represented by a rectangle.

of G is assigned a weight d_{ij} equal to the Manhattan length of the *shortest* leader, which connects area-site s_i to label l_j . According to Theorem 5.1, this step costs $O(n^2 \log n)$ time, assuming that the maximum number of corners of an area-site is fixed.

In Step B of Algorithm 5.1, we have to compute a minimum-cost bipartite matching on the output graph of Algorithm 5.1. As already mentioned this will cost $O(n^3)$ time. The solution obtained from Step C of Algorithm 5.1 might contain crossing leaders. In Step D of Algorithm 5.1, the crossings are eliminated in $O(n \log n)$ time. Thus, the total time complexity of Algorithm 5.1 is $O(n^3)$. \square

5.5. Sample labeling of type-*opo* Leaders

Figures 10 and 11 depict the regions of Germany. In both figures, the labels are placed on two opposite sides of R and the leaders are of type *opo*. In Fig. 10, a point is arbitrarily selected as the representative of each region. The labeling of Fig. 10 is visually improved in Fig. 11 by replacing these points with rectangles within each region. Both labelings are optimal in terms of total leader length. However, the total leader length of Fig. 11 is reduced by 37% compared to that of Fig. 10. Note that we also achieved the reduction in the number of leader bends from 8 (in Fig. 10) to 5 (in Fig. 11) just by the use of rectangular area-sites instead of point-sites.

6. TWO-SIDED AREA-FEATURE BOUNDARY LABELINGS WITH TYPE-*po* LEADERS

In this section, we study the area-feature boundary labeling problem with leaders of type *po*. We further assume that we have uniform-sized labels with sliding label-ports, placed on two opposite sides of the enclosing rectangle R . Our objective is to determine a legal boundary labeling so that the total leader length is minimized.

To deal with this problem, we use the (matching based) Algorithm 5.1 for the case of type-*opo* leaders to get a labeling of minimum total leader length. This can be done in $O(n^3)$ time. We proceed by replacing the type-*opo* leaders with type-*po* leaders. Note that connecting a site to its label with a type-*opo* or a type-*po* leader requires leaders of the same length under the Manhattan metric, assuming that we keep the positions of both site and label-ports unchanged. Therefore, the solution obtained in this manner remains optimal in terms of total leader length, but it might contain crossings.

Possible crossings between leaders to the same side of the enclosing rectangle are resolved following a similar strategy as the one presented for the corresponding point-site labeling problem [4], without changing the total leader length.³ The main idea

³Alternatively, we can use the $O(n \log n)$ algorithm of Benkert and Nöllenburg [8] to solve the *po*-labeling problem. For completeness, we prefer to present a simpler $O(n^2)$ algorithm, since the higher time complexity of this step does not affect the overall $O(n^3)$ time complexity of the algorithm.

of this approach is to consider the sites from top to bottom and by performing appropriate reroutings to eliminate all crossings. The i -th site in this order may require at most $i - 1$ reroutings. Hence, the crossing elimination procedure can be done in $O(n^2)$ additional time [4, p. 229]. Moreover, we can easily observe that crossings between leaders that go to opposite sides of the enclosing rectangle cannot occur. This is due to the fact that swapping these crossing leaders would result in a solution with smaller total leader length, which is a contradiction since we assume that the original solution minimizes the total leader length. Note that the same strategy can be applied in the case where the labels can be placed on only one side of the enclosing rectangle R . The following theorem summarizes our result.

THEOREM 6.1. *Given a label set L of n uniform-sized labels placed at fixed positions along two opposite sides of a rectangle R and a site set P of n area-sites in general position inside R , we can compute in $O(n^3)$ time a legal po -labeling of minimum total leader length.*

For the case where the labels occupy two adjacent sides of the enclosing rectangle R , there exist instances of the problem where it is not feasible to determine a crossing-free type- po labeling, irrespective of the labeling optimization criteria (in contrast to the case where labels occupy only opposite sides; refer to Theorem 6.1). This follows from the observation that the rerouting of two crossing po -leaders does not always eliminate their crossing (see Fig. 12). Figure 12c illustrates an instance of the problem where the labels occupy all four sides of R . All sites are contained within the gray-colored rectangular area that is formed by (i) the top left corner of the enclosing rectangle R and (ii) the intersection of the lines that coincide with the bottom side of the topmost label on the left side of R and the right side of the leftmost label on the top side of R . Since two sites have to be connected to the dark-gray-colored labels (incident to the bottom-right corner of R), a crossing as the one of Fig. 12 will inevitably arise, which implies that this instance admits no crossing-free solution. The following observation summarizes this result.

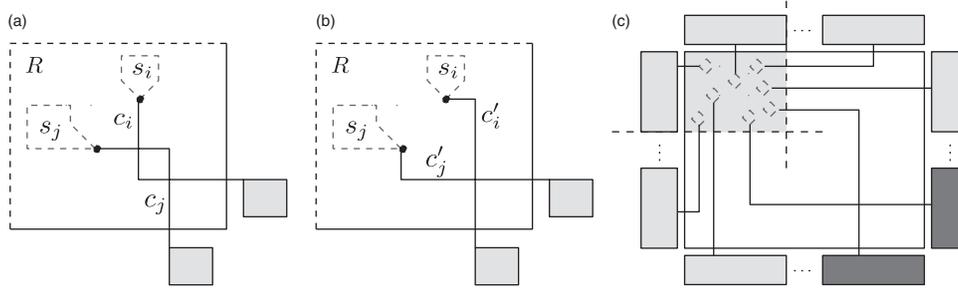


FIGURE 12. Two crossing po -leaders does not always eliminate their crossing: (a) before rerouting; and (b) after rerouting; (c) an four-sided labeling instance which admits no crossing-free.

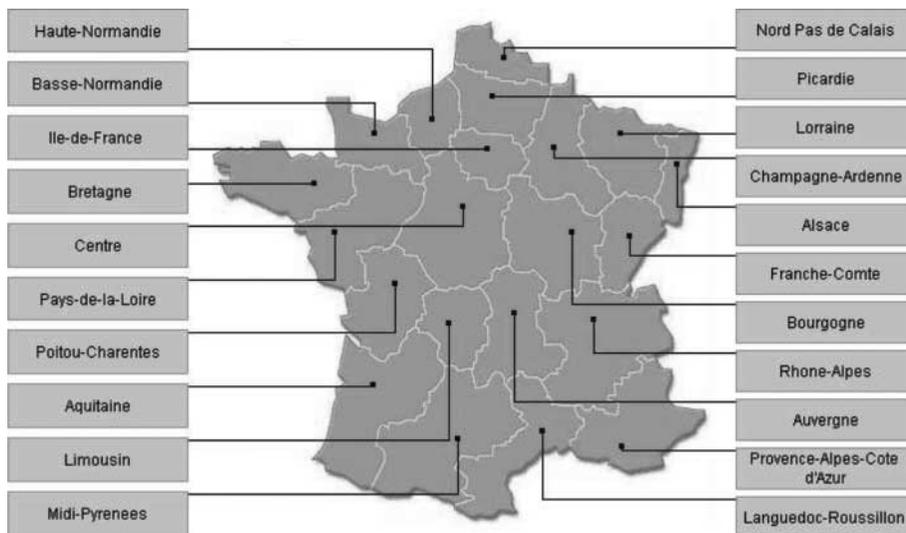


FIGURE 13. A po -labeling of France; a point that geographically coincides with the capital of each region is taken as its representative.

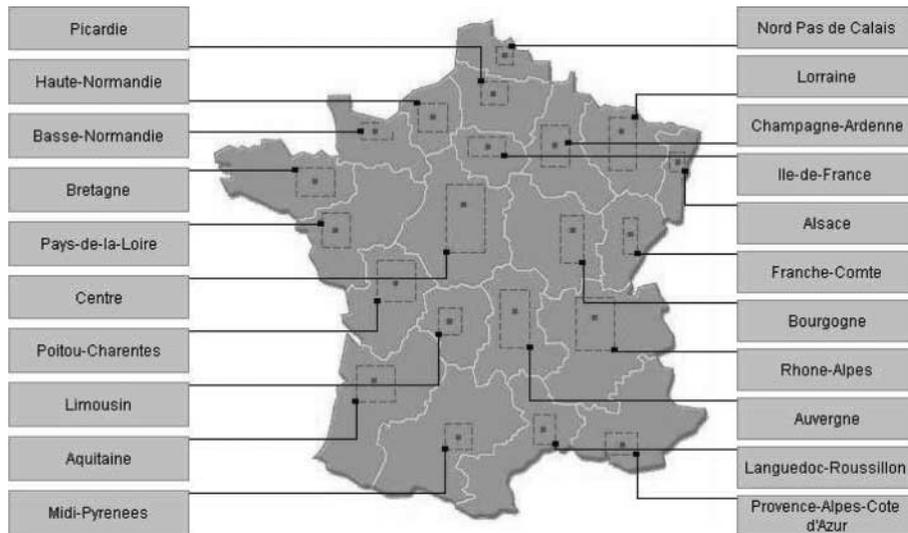


FIGURE 14. A visually improved map for the drawing of Fig. 13; each region is represented by a rectangle.

OBSERVATION 6.1. Consider the area-feature boundary labeling problem with type- po leaders and uniform labels placed at fixed positions. If labels occupy two adjacent sides of the enclosing rectangle, there exist instances that admit no legal labeling irrespective of the labeling optimization criteria.

6.1. Sample labeling of type- po leaders

Figures 13 and 14 depict the regions of France. In both figures, the labels are restricted to two opposite sides of R and the leaders are of type po . In Fig. 13, a point that geographically coincides with the capital of each region is selected as the region's representative. The labeling of Fig. 13 is visually improved in Fig. 14 by replacing these points with small rectangles that enclose the capitals and totally lie within each region. Both labelings are optimal in terms of total leader length. However, the total leader length of Fig. 14 is reduced by 11.5% compared to that of Fig. 13. Note that in Fig. 14 the vertical order of the sites that are mapped to labels at the left (right) side of R is nearly the same as the corresponding order of the labels on the left (right) side of R , in contrast to Fig. 13. We also achieved the reduction of the number of leader bends from 16 (in Fig. 13) to 12 (in Fig. 14), just by the use of rectangular area-sites instead of points. It should be pointed out that the purpose of the provided sample drawings is to demonstrate that the use of rectangles (that is, of one of the simplest types of area-sites) to represent sites can result in a drawing of better quality. Such an improved drawing cannot always be obtained by simply moving the point representing each region to the bend-point of its corresponding leader in the optimal solution. This is due to the fact that several of the bend-points lie outside the region of their corresponding leader (this is the case for six out of the twenty one sites in Fig. 13).

7. CONCLUSIONS

In this paper, we considered the area-feature boundary labeling problem where the sites are presented as simple polygons in general position instead of points. We presented an efficient algorithm to determine opo -labelings of minimum total leader length, where the labels are of uniform size with sliding label-ports, placed on all four sides of the enclosing rectangle R (the cases where labels are placed on one, two or three sides only, are all handled as subcases of the four-side setting). For the special case, where the labels are placed on two opposite sides of the enclosing rectangle R , we extended this algorithm to support po -leaders (the case where labels are placed on only one side, is handled as a subcase of the two opposite-side setting).

All known type- opo boundary labelings use a track-routing area outside the enclosing rectangle to route the p -segment of the leader, where its final o -segment to the labels is only necessary to avoid otherwise overlapping p -segments. The main drawback of this approach occurs when a large number of p -segments lie within the track-routing area without an appropriate distance among them. The problem becomes more significant if we also take into account the bends that each leader contributes and that are all placed in the track-routing area. Routing the p -segment inside the enclosing rectangle might lead to visually improved labelings. The theoretical study of a mixed model where we permit combinations of different types of leaders would also be interesting.

The evaluation of different optimization criteria (e.g. the one of minimizing the total number of leader bends) would also be of particular interest. Furthermore, no results were presented for the cases of sliding labels or non-uniform labels. For the latter problem very few results exist in boundary labeling, in general.

Another line of research would be to try to find efficient ways of determining for each region of a map a representative simple polygon that has less than k corners, where k is a constant.

FUNDING

This work is partially funded by the project PENED 2003. The project is co-funded by the European Social Fund (75%) and National Resources (25%).

REFERENCES

- [1] Bekos, M.A., Kaufmann, M., Potika, K. and Symvonis, A. (2006) Polygons Labelling of Minimum Leader Length. In Kazuo, M., Kozo, S. and Jiro, T. (eds.), *Proc. 5th Asia Pacific Symp. Information Visualisation (APVIS2006)*, Tokyo, Japan, February 1–3, CRPIT, Vol. **60**, pp. 15–21. Australian Computer Society. ISBN: 1-920682-41-4.
- [2] Formann, M. and Wagner, F. (1991) A Packing Problem with Applications to Lettering of Maps. *Proc. 7th Annual ACM Symp. Computational Geometry*, North Conway, NH, USA, pp. 281–288.
- [3] Wolff, A. and Strijk, T. The map-labeling bibliography. <http://i11www.ira.uka.de/map-labeling/bibliography>.
- [4] Bekos, M.A., Kaufmann, M., Symvonis, A., and Wolff, A. (2007) Boundary labeling: models and efficient algorithms for rectangular maps. *Computational Geom., Theory Appl.*, **36**, 215–236.
- [5] Vaidya, P.M. (1989) Geometry helps in matching. *SIAM J. Comput.*, **18**, 1201–1225.
- [6] Bekos, M.A., Kaufmann, M., Potika, K. and Symvonis, A. (2006) Multi-stack Boundary Labeling Problems. In Arun-Kumar, S. and Garg, N. (eds.), *Proc. 26th Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS2006)*, Kolkata, India, December 13–15, Lecture Notes in Computer Science 4337, pp. 81–92. Springer.
- [7] Bekos, M.A., Kaufmann, M. and Symvonis, A. (2007) Labeling Collinear Sites. In Hong, S.-H. and Ma, K.-L. (eds.), *Proc. 6th Int. Asia-Pacific Symp. Visualization (APVIS2007)*, Sydney, Australia, February 5–7, pp. 45–51. IEEE.
- [8] Benkert, M. and Nöllenburg, M. (2007) Improved Algorithms for Length-Minimal One-Sided Boundary Labeling. *Proc. 23rd European Workshop on Computational Geometry (EWCG'07)*, Graz, Austria, March 19–21, pp. 190–193.
- [9] Benkert, M., Haverkort, H., Kroll, M. and Nöllenburg, M. (2007) Algorithms for Multi-criteria One-Sided Boundary Labeling. In Hong, S.-H., Nishizeki, T. and Quan, W. (eds.), *Proc. 15th Int. Symp. Graph Drawing (GD'07)*, Sydney, Australia, September 24–26, Lecture Notes in Computer Science 4875, pp. 243–254. Springer.
- [10] Bekos, M.A., Kaufmann, M., Nöllenburg, M. and Symvonis, A. (2008) Boundary Labeling with Octilinear Leaders. In Gudmundsson, J. (ed.), *Proc. 11th Scandinavian Workshop on Algorithm Theory (SWAT08)*, Gothenburg, Sweden, July 2–4, Lecture Notes in Computer Science 5124, pp. 234–245. Springer.
- [11] Kao, H.-J., Lin, C.-C. and Yen, H.-C. (2007) Many-to-One Boundary Labeling. In Hong, S.-H. and Ma, K.-L. (eds.), *Proc. 6th Int. Asia-Pacific Symp. Visualization (APVIS2007)*, Sydney, Australia, February 5–7, pp. 65–72. IEEE.
- [12] de Berg, M., van Kreveld, M., Overmars, M. and Schwarzkopf, O. (2000) *Computational Geometry: Algorithms and Applications*, (2nd edn). Springer.
- [13] Lee, D.T. (1980) Two-dimensional Voronoi diagrams in the LP-metric. *J. Assoc. Comput. Mach.*, **27**, 604–618.
- [14] Kuhn, H.W. (1955) The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.*, **2**, 83–97.
- [15] Mehlhorn, K. (1984) *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. Springer.